Coupling non-conforming NURBS patches in isogeometric analysis of complex thin shell structures

L. Coox[1,2], **F. Greco**[1,2], **F. Maurin**[1,2], **D. Vandepitte**[1,2], **W. Desmet**[1,2]
[1] KU Leuven, Department of Mechanical Engineering,
Celestijnenlaan 300, B-3001, Leuven, Belgium
e-mail: **laurens.coox@kuleuven.be**

[2] Member of Flanders Make

Abstract
Isogeometric analysis allows the use of shell formulations without rotational degrees of freedom, but describing complex geometries typically requires multiple NURBS patches, that are often non-conforming. This work presents a flexible method for coupling such multipatch geometries in isogeometric frameworks and applies it to rotationless Kirchhoff-Love shell analysis. The rotationless formulation requires the enforcing of C^1-continuity across patch interfaces. The present work does this by satisfying interface constraints in a master-slave formulation, where the interface constraints can be derived using only the mesh information. Eliminating the slave variables from the system of equations results in a reduced system matrix. Whereas the C^0-part of the coupling is a global coupling in the weak form, the C^1-continuity is enforced by a strong point-wise coupling in certain well-chosen collocation points along the interface. The proposed method can be applied without any user interaction, both for conforming and for non-conforming patch configurations. This is demonstrated for a set of problems of dynamic shell analysis. Together with the rotationless character of the isogeometric shell formulation, the coupling method allows analysing complex shell structures in a computationally efficient manner.

1 Introduction

IsoGeometric Analysis (IGA) is a numerical modelling method introduced by Hughes et al. [1, 2]. It can be seen as a generalisation of the Finite Element Method (FEM) that uses spline-based shape functions. In doing so, IGA addresses the issue that geometry descriptions in Computer Aided Design (CAD) on the one hand and in Computer Aided Analysis (CAE) on the other hand remain very different, even though they represent the same geometry. This difference requires time-consuming meshing steps to translate a CAD geometry to an analysis-suitable CAE mesh. IGA therefore aims to unify the geometry representations in design and analysis by introducing CAD basis functions into a CAE environment. The traditional element-based discretisation and the associated (usually low-order polynomial) shape function expansions are replaced by CAD-based mappings and associated functions, which are typically spline-based descriptions like Non-Uniform Rational B-Splines (NURBS). This allows the CAD geometry to be directly plugged into the CAE tool, in essence rendering the meshing step redundant. IGA has already been studied extensively and applied successfully to a multitude of problem types, as illustrated by the overview presented in [3].

A particularly interesting application where NURBS basis functions can add a lot of value is thin shell analysis. For thin shells, transverse shear deformations can be neglected, making the Kirchhoff-Love theory applicable. This in contrast to the Reissner-Mindlin theory, which does take into account those shear deformations and is typically used for thick shell analysis. But although a lot of shell structures in engineering practice can be regarded as thin shells, the use of Reissner-Mindlin theory is significantly more common in

state-of-the-use finite element codes. This is mainly because Reissner-Mindlin elements only require C^0-continuity, whereas the Kirchhoff-Love theory needs C^1-continuous elements. The latter is not so easy to obtain for arbitrary geometries when using conventional Lagrange polynomials as shape functions. NURBS functions, on the other hand, are much smoother and can easily achieve higher inter-element continuities. Isogeometric shell elements are therefore very appealing for use in a Kirchhoff-Love shell formulation. This avoids the use of rotational Degrees Of Freedom (DOFs) — thereby significantly reducing the resulting system matrix size — and in addition does not suffer from the locking phenomena typically encountered in (low-order) Reissner-Mindlin elements. Moreover, the geometrically exact character of the isogeometric approach can be a crucial advantage for shell models, which can be very sensitive to small imperfections [4].

Despite the fast progress IGA has made since its conception in 2005 [1], its application in a NURBS-based context still presents some open issues. These mainly stem from the tensor-product topology of NURBS, which makes free-form geometries difficult to create using a single NURBS surface patch. Complex geometries therefore usually consist of several patches. Although C^1-continuity is easily achieved within a single patch, it is not straightforward to obtain across patch boundaries — even maintaining C^0-continuity can be challenging for arbitrary patch configurations. Given their importance in IGA, patch coupling methods have already been the subject of many research papers — although mainly focused on C^0-coupling [5–10]. Typically, mortar methods [9, 10], Nitsche's method [7, 8], or penalty approaches [8] provide the coupling. The C^1-coupling of conforming patches has been studied particularly in the context of Kirchhoff-Love shells [11–15]. Maurin et al. [15] present a C^1-coupling approach in an isogeometric framework for unit cell modelling. The bending strip method proposed by Kiendl et al. in [12] couples two bordering patches by adding a strip of fictitious material with a unidirectional bending stiffness, thus preventing patch interfaces from acting as hinges. Schmidt et al. [13] employ this technique for coupling trimmed NURBS patches. This bending strip method, however, is rather limited in the interface types it can handle. The presence of non-conforming patches, for instance, further complicates the patch coupling, but is very common in typical CAD geometries. The modelling of complex shell structures therefore requires a framework that can impose C^1-continuity of non-conforming patches in a flexible manner. The method presented in [16] introduces a robust framework for coupling non-conforming patches based on fundamental NURBS refinement properties. Although it mainly focuses on C^0-problems [17], it also illustrates a simple method of collinear control variables for enforcing C^1-continuity for relatively simple geometries. Breitenberger et al. [18] also treat the C^1-coupling of Kirchhoff-Love shells (including non-conforming and even trimmed geometries), but by means of a penalty approach.

This work introduces a method for coupling non-conforming NURBS patches in Kirchhoff-Love shell analysis in a flexible and user-friendly manner, without requiring the determining of proper penalty factors or introducing additional variables like Lagrange multipliers or rotational DOFs at the patch boundaries. The remainder of the text is structured as follows. The next section briefly introduces some preliminaries on NURBS and their use in IGA. Section 3 then presents the proposed coupling method, detailing both the C^0- and the C^1-coupling parts. Its accuracy is then illustrated through numerical case studies in section 4, followed by some concluding remarks.

2 Isogeometric analysis in a NURBS-based context

This section introduces the terminology and formulae regarding NURBS surfaces and the use of their corresponding basis functions as shape functions in the context of IGA. A more detailed explanation about IGA can be found in [1, 2] and about NURBS in particular in [19].

NURBS are built starting from B-splines, which are piecewise polynomial functions. A set of B-splines is defined by a knot vector $\boldsymbol{\Xi} = [\xi_1, \xi_2, \ldots, \xi_{n+p+1}]$ for a given polynomial order p. This knot vector is a sequence of non-decreasing coordinates in the parameter space ξ, where $\xi_k \in \mathbb{R}$ $(k = 1, 2, \ldots, n + p + 1)$ with n the number of basis functions making up the B-spline. Open knot vectors are standard in CAD, with the first and the last knot each having a multiplicity of $p + 1$. A set of n one-dimensional B-spline

basis functions $N_i^p(\xi)$ ($i = 1, \ldots, n$) can then be obtained recursively using the Cox – de Boor recursion formula [20, 21]:

$$\begin{aligned} p = 0: \quad & N_i^0(\xi) = \begin{cases} 1 & \xi_i \leq \xi < \xi_{i+1} \\ 0 & \text{otherwise} \end{cases}, \\ p > 0: \quad & N_i^p(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_i^{p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1}^{p-1}(\xi). \end{aligned} \tag{1}$$

To now generate a NURBS basis $R_i^p(\xi)$, a weight w_i is assigned to every B-spline function $N_i^p(\xi)$:

$$R_i^p(\xi) = \frac{N_i^p(\xi) w_i}{W(\xi)} = \frac{N_i^p(\xi) w_i}{\sum_{i_w=1}^{n} N_{i_w}^p(\xi) w_{i_w}} \quad \text{for } i = 1, \ldots, n, \tag{2}$$

with $W(\xi)$ being the weighting function. Each $R_i^p(\xi)$ is a piecewise rational function, since $N_i^p(\xi)$ and $W(\xi)$ are both piecewise polynomial. In fact, B-splines are a subset of NURBS that are obtained when all the weights w_i are equal. It is worth noting that NURBS bases form partitions of unity. Using a set of vector-valued weighting coefficients $\mathbf{P}_i \in \mathbb{R}^3$ called control points, the basis functions can be combined to form a NURBS curve $\mathbf{C}(\xi)$:

$$\mathbf{C}(\xi) = \sum_{i=1}^{n} R_i^p(\xi) \mathbf{P}_i. \tag{3}$$

Multivariate NURBS entities are generated by taking the tensor product of univariate NURBS bases. Given a pair of knot vectors $\boldsymbol{\Xi} = [\xi_1, \xi_2, \ldots, \xi_{n+p+1}]$ and $\boldsymbol{H} = [\eta_1, \eta_2, \ldots, \eta_{m+q+1}]$ for two polynomial orders p and q, a bivariate NURBS basis is defined as

$$R_{i,j}^{p,q}(\xi, \eta) = \frac{N_i^p(\xi) M_j^q(\eta) w_{i,j}}{\sum_{i_w=1}^{n} \sum_{j_w=1}^{m} N_{i_w}^p(\xi) M_{j_w}^q(\eta) w_{i_w, j_w}} \quad \text{for } \begin{matrix} i = 1, \ldots, n \\ j = 1, \ldots, m \end{matrix}, \tag{4}$$

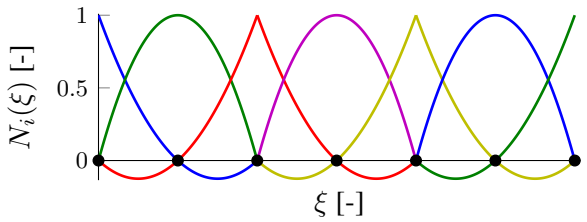
where $N_i^p(\xi)$ and $M_j^q(\eta)$ represent univariate B-spline basis functions of respectively order p and q, associated with knot vectors $\boldsymbol{\Xi}$ and $\boldsymbol{H}$, respectively. A NURBS surface geometry $\boldsymbol{S}(\xi, \eta)$ is then defined in analogy with a NURBS curve using a net of control points $\mathbf{P}_{i,j} \in \mathbb{R}^3$:

$$\mathbf{S}(\xi, \eta) = \sum_{i=1}^{n} \sum_{j=1}^{m} R_{i,j}^{p,q}(\xi, \eta) \mathbf{P}_{i,j}. \tag{5}$$

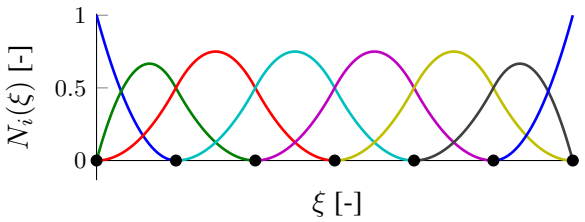
It should be noted that the control points are in general not interpolatory, meaning that they do not necessarily lie on the surface they define. Figure 1 illustrates the differences between conventional polynomials and NURBS when used for quadratic shape functions in the one-dimensional case. The figure clearly shows that the Lagrangian functions are interpolatory but only C^0-continuous. In contrast to this, quadratic NURBS are C^1-continuous but do not interpolate the nodes. They also have a wider support than the Lagrangian polynomials, although this does not lead to system matrices with a higher bandwidth. The total number of functions that any of the basis functions shares support with is at most $2p + 1$, for NURBS as well as for Lagrangian polynomials. It is also worth noting that the quadratic Lagrange elements all require a middle node, meaning that the 7 nodes in fig. 1a represent 3 elements. In the NURBS case, each node (i.e. unique knot) represents an element boundary, resulting in 6 elements in fig. 1b. In general, for a given polynomial order and number of elements, a NURBS mesh contains fewer shape functions than a conventional polynomial mesh.

Applying the isoparametric paradigm, a NURBS basis can be used in shape function expansions to approximate field variables. Taking $u(\xi, \eta)$ to represent the (scalar) primary variable, its approximation $\hat{u}(\xi, \eta)$ becomes:

$$u(\xi, \eta) \approx \hat{u}(\xi, \eta) = \sum_{i=1}^{n} \sum_{j=1}^{m} R_{i,j}^{p,q}(\xi, \eta) \cdot d_{i,j} = \sum_{l=1}^{n \cdot m} R_l^{p,q}(\xi, \eta) \cdot d_l, \tag{6}$$



(a) Quadratic Lagrangian shape functions for equally spaced nodes.



(b) Quadratic NURBS shape functions for a uniform open knot vector. These NURBS functions are B-splines $N_i^2(\xi)$ as defined in eq. (1).

Figure 1: One-dimensional quadratic shape functions for a mesh consisting of seven nodes (indicated by •) using (a) conventional polynomials and (b) NURBS.

where d_l are the unknown control variables to be solved for, with $l = i + (j-1)\cdot m$ a single counter replacing i and j. Using a Galerkin approach, these same shape functions are then also used as test functions in a weighted residual formulation.

Equation (5) represents a single NURBS surface patch. A general NURBS surface, however, typically consists of a combination of multiple patches — so-called *multipatch* surfaces. Each patch is formed by a concatenation of knot spans. These are bounded by (non-identical) knots and represent the elements in the mesh. Across a knot ξ_i, the basis functions have a C^{p-m_i}-continuity, with m_i the multiplicity of knot ξ_i. NURBS bases are therefore very flexible regarding inter-element continuity: almost arbitrarily high continuities can be obtained. Across patch boundaries, however, care needs to be taken to ensure that the field variable space maintains the required continuity. When enforcing C^0-continuity across patches with matching meshes, this process is trivial, but it cannot be done straightforwardly in a more general case.

3 Multipatch isogeometric Kirchhoff-Love shell structures

The shell structures in this work are modelled using isogeometric Kirchhoff-Love shells [11], taking into account membrane and bending effects while neglecting transverse shear deformations. Consider a thin, flat shell of thickness t (much smaller than its other dimensions), and of a homogeneous, isotropic and linearly elastic material of density ρ, Young's modulus E and Poisson's coefficient ν. Let $\boldsymbol{X} = \{X_1, X_2, X_3\}^T$ be the undeformed configuration in three-dimensional space, with X_3 the out-of-plane coordinate of the cartesian coordinate system. The deformation $\boldsymbol{u} = \{u_1, u_2, u_3\}^T$ of the midsurface then fully describes the shell deformation. Assuming small deformations, the in-plane and out-of-plane motions are decoupled and are governed by separate dynamic equations. The steady-state dynamic equation of bending in a thin flat shell (using a harmonic time-dependency given by the angular frequency ω) is governed by the following equation [22]:

$$\boldsymbol{\nabla}^4 u_3(X_1, X_2) - k_b^4 \cdot u_3(X_1, X_2) = \frac{p(X_1, X_2)}{D}, \tag{7}$$

with $\boldsymbol{\nabla}^4 \bullet = \frac{\partial^4 \bullet}{\partial X_1^4} + \frac{\partial^4 \bullet}{\partial X_1^2 \partial X_2^2} + \frac{\partial^4 \bullet}{\partial X_2^4}$ the biharmonic operator and $p(X_1, X_2)$ the external (normal) load. The bending stiffness D and the bending wavenumber k_b are defined as:

$$D = \frac{Et^3}{12(1-\nu^2)}, \tag{8}$$

$$k_b = \sqrt[4]{\frac{\rho_s t \omega^2}{D}}. \tag{9}$$

The in-plane motion is given by $u_1(X_1, X_2)$ and $u_2(X_1, X_2)$ and is governed by the following dynamic equations of motion (leaving out the (X_1, X_2)-dependency for the sake of brevity):

$$\begin{cases} \dfrac{\partial^2 u_1}{\partial X_1^2} + \dfrac{1-\nu}{2}\dfrac{\partial^2 u_1}{\partial X_2^2} + \dfrac{1+\nu}{2}\dfrac{\partial^2 u_2}{\partial X_1 \partial X_2} + \dfrac{1-\nu^2}{E} b_{X_1} + \dfrac{\rho(1-\nu^2)\omega^2}{E} u_1 = 0 \\ \dfrac{\partial^2 u_2}{\partial X_2^2} + \dfrac{1-\nu}{2}\dfrac{\partial^2 u_2}{\partial X_1^2} + \dfrac{1+\nu}{2}\dfrac{\partial^2 u_1}{\partial X_1 \partial X_2} + \dfrac{1-\nu^2}{E} b_{X_2} + \dfrac{\rho(1-\nu^2)\omega^2}{E} u_2 = 0 \end{cases}, \tag{10}$$

with b_{X_i} the body force acting on the shell in direction X_i.

For curved shells, the bending and membrane phenomena are no longer decoupled like in the flat case, but by using convective curvilinear coordinates the stresses and strains in the curved shell can be separated into bending and membrane contributions. For details on this and on other aspects concerning the employed Kirchoff-Love shell theory, the interested reader is referred to [11]. The present work studies dynamic, time-harmonic problems which means that the formulation in [11] needs to be extended to also take into account inertia effects (represented by the terms in ω^2 in eqs. (7) and (10)).

Equation (10) is a set of second-order partial differential equations, and a weighted residual formulation of the weak form therefore requires C^0-continuous elements. Equation (7), however, is of fourth order, meaning that the shape function compatibility requirements demand at least C^1-continuity. In traditional shell elements, with C^0-continuity between them, this is typically ensured by introducing rotational DOFs. An isogeometric approach on the other hand, with higher continuity of the NURBS basis functions, allows the implementation of rotationless shells using only the translational displacements as DOFs (in this case the deformation in a global coordinate system). The C^1-continuity is embedded in the NURBS basis, provided that quadratic or higher-order functions are used. This holds true within a single NURBS patch, but across patch borders the C^1- and even the C^0-continuity needs to be explicitly enforced. The remainder of this section describes how this is done in the present work for the general case of non-conforming meshes.

3.1 Enforcing C^0-continuity across patch boundaries

For conforming meshes (i.e. meshes with matching mesh lines and the same interface basis functions), the C^0-coupling is trivial and can be done by simply merging corresponding control variables on the interface (i.e. matching them one-to-one). This is not an option for non-conforming meshes, in which case the technique explained below is used; a more detailed study of it can be found in [16].

Consider two non-conforming patches like in the top left illustration of fig. 2. They are numbered 1 and 2, and are assumed to be of the same degree. To couple them, they are virtually refined until they are conforming (like in the top right of the figure), after which the regular one-to-one matching is enforced. This virtual refinement is done by inserting into the interface knot vectors the uncommon knots: the knots that are present in the other patch's knot vector, but not in that of the patch itself. In other words: given the two interface knot vectors $\boldsymbol{\Xi}_1$ and $\boldsymbol{\Xi}_2$, refine them until they are both identical to $\boldsymbol{\Xi}_{1\cup 2} = \boldsymbol{\Xi}_1 \cup \boldsymbol{\Xi}_2$ (taking into account knot multiplicities). This refinement process is referred to here as *virtual uncommon knot insertion*. It is virtual in the sense that the refined discretisations are never used for shape function expansions; they are only used for generating the coupling equations resulting from a one-to-one matching of the refined control variables. It can be represented by extraction operators A_1 and A_2 (similar to those for Bézier extraction presented in [23]) that link the original control variables $\mathbf{d}_1$ and $\mathbf{d}_2$ to the control variables $\mathbf{d}_1^v$ and $\mathbf{d}_2^v$ of the virtually refined versions:

$$\mathbf{d}_i^v = \mathsf{A}_i \mathbf{d}_i \qquad i = 1, 2. \tag{11}$$

These extraction operators A_i contain the linear combinations of the original control points that are required to obtain each refined control point. Details on how to compute them are given in [16], but it is worth remarking that their assembly only requires information about the knot vector and about the NURBS weights.

No shape function evaluations whatsoever are needed. Ensuring the continuity now consists in enforcing $\mathbf{d}_1^v = \mathbf{d}_2^v$, or equivalently, using the original control variables $\mathbf{d}_i$:

$$\mathsf{A}_1\mathbf{d}_1 = \mathsf{A}_2\mathbf{d}_2. \tag{12}$$

Directly enforcing this entire set of constraints results in exact satisfaction of the C^0-continuity but will in general lead to an overconstraining of the interface and convergence to the wrong global solution. Therefore, both patches are instead coupled in a weak sense by expressing the variables of the more finely discretised patch (the slave) in function of those of the more coarse one (the master). Using subscripts 'm' and 's' for the master and slave patch, respectively, this yields the following continuity relationship:

$$\mathbf{d}_\mathrm{s} = \mathsf{A}_\mathrm{s}^+\mathsf{A}_\mathrm{m}\mathbf{d}_\mathrm{m}, \tag{13}$$

with A_s^+ the pseudo-inverse of A_s. This reduces the number of interface constraints by satisfying eq. (12) in a least-squares sense, allowing an accurate global solution without overconstraining the model. It is worth noting that in the special case of hierarchical meshes (where one patch is a refined version of the other one), eq. (13) leads to exact C^0-continuity across the entire interface (since in that case A_s is the identity matrix).

Figure 2 illustrates this coupling procedure. The bottom part of the figure represents the coupled patches (with $\mathbf{d}_\mathrm{s}$ and $\mathbf{d}_\mathrm{m}$ linked by eq. (13)), the top left part the uncoupled ones. The top right illustration shows the refined (virtual) configuration, where $\mathbf{d}_1^v = \mathbf{d}_2^v$ has to hold.

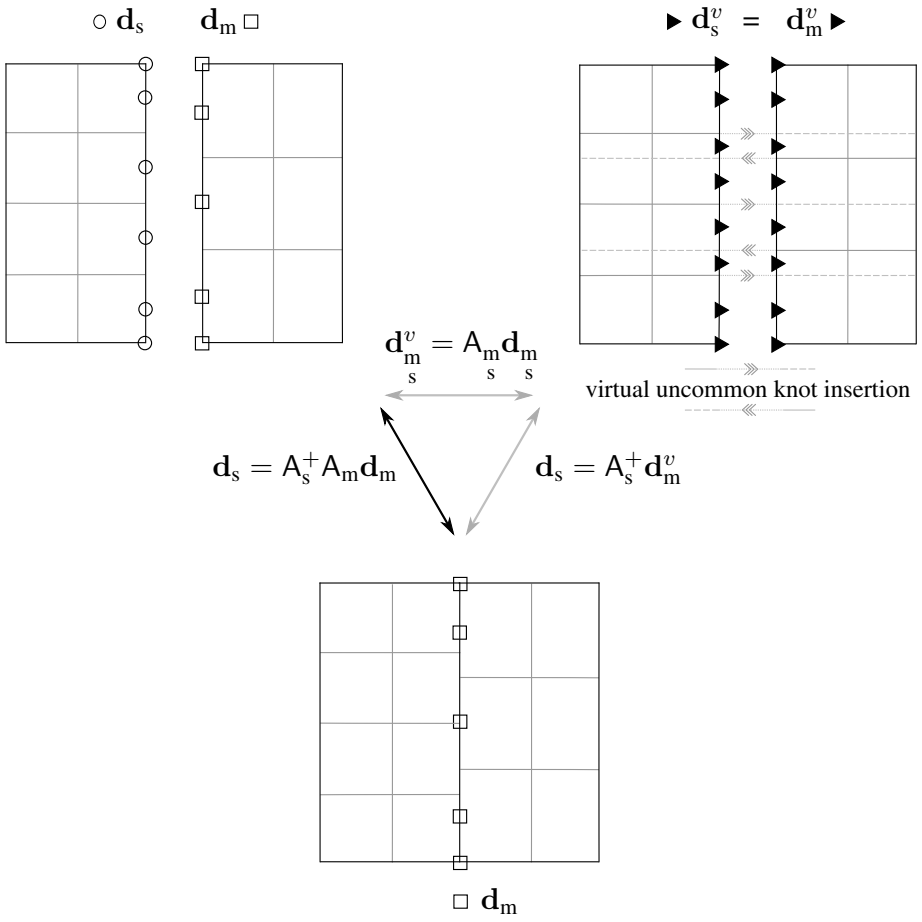


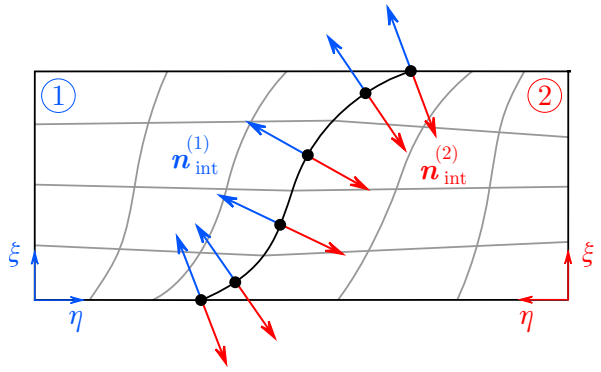
Figure 2: Illustration of the C^0 patch coupling concept. Using restrictions resulting from the one-to-one matching of (virtually) uncommon knot-inserted meshes (top right) that are transformed back to the original control variables (top left), C^0-continuity conditions can be (weakly) enforced within a master-slave relationship (bottom), only retaining the master DOFs. Because the right patch is the less refined one, it acts as the master.

3.2 Adding C^1-continuity across patch boundaries

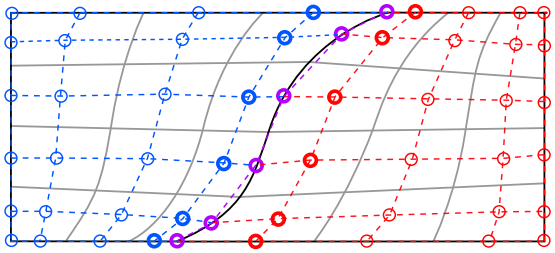
Consider two conforming patches with an arbitrary interface curve like in fig. 3a. Without loss of generality, the interface curve is assumed to be the patch boundary located at $\eta = 1$ for both patches, meaning the parametric coordinate running along the interface is ξ. Along this parametric coordinate (i.e. tangential to the interface), the discretisation is already C^1-continuous. To now ensure the C^1-continuity of the out-of-plane displacement $u_3(\xi, \eta)$ across the interface, the following should hold:

$$\frac{\partial u_3^{(1)}}{\partial n_{\text{int}}^{(1)}}(\xi, 1) + \frac{\partial u_3^{(2)}}{\partial n_{\text{int}}^{(2)}}(\xi, 1) = 0, \tag{14}$$

where the superscript $\bullet^{(i)}$ refers to quantities pertaining to patch i, and with $\boldsymbol{n}_{\text{int}}^{(i)}(\xi)$ being the in-plane unit vector normal to the interface and pointing towards the interior of the patch (cf. fig. 3a). It can be calculated as the normalised version of the vector product $\frac{\partial \mathbf{S}^{(i)}(\xi,\eta)}{\partial \xi} \times \left(\frac{\partial \mathbf{S}^{(i)}(\xi,\eta)}{\partial \xi} \times \frac{\partial \mathbf{S}^{(i)}(\xi,\eta)}{\partial \eta} \right)$. Imposing constraint (14) is significantly less straightforward than enforcing the purely C^0 coupling. This is mainly because the shape function derivatives on the interface are not only influenced by the interface control variables, but also by those next to the interface. The derivatives therefore strongly depend on the specific geometrical mapping, which prevents an elegant coupling that requires only the linking of control variables without the need for shape function evaluations — except in some special cases like the ones presented in [16]. Therefore, a collocation approach is chosen to enforce eq. (14) point-wise.



(a) The two NURBS surface $\mathbf{S}^{(i)}(\xi, \eta)$ with an indication of the collocation points (•) and the interface normals $\boldsymbol{n}_{\text{int}}$.



(b) Overlay of the mesh with the control polygon. The control variables indicated in bold determine the shape function derivatives at the interface.

Figure 3: A (planar) conforming two-patch configuration illustrating the C^1 coupling procedure. The grey lines delineate the elements.

The images of Greville abscissae [24] are used as collocation points. Greville abscissae are computed as knots averages, and their interpolation is proved to be stable (in 1D) up to at least polynomial order 3 [25]. They have already been applied successfully in isogeometric collocation frameworks, but there are also alternatives like Demko abscissae [26]. These are proved to be always stable, but need to be obtained iteratively. Given a knot vector $[\xi_1, \xi_2, \ldots, \xi_{n+p+1}]$, the n Greville abscissae $\bar{\xi}_i$ are obtained as

$$\bar{\xi}_j = \frac{1}{p}\left(\xi_{j+1} + \xi_{j+2} + \ldots + \xi_{j+p}\right), \quad j = 1, \ldots, n. \tag{15}$$

This means that there are as many collocation points as there are basis functions (in a single patch) along the interface, namely n. Collocating eq. (14) in (the images of) these Greville abscissae then yields

$$\frac{\partial u_3^{(1)}}{\partial n_{\text{int}}^{(1)}}(\bar{\xi}_j, 1) + \frac{\partial u_3^{(2)}}{\partial n_{\text{int}}^{(2)}}(\bar{\xi}_j, 1) = 0, \quad j = 1, \ldots, n. \tag{16}$$

Or, considering a NURBS shape function expansion for u_3 like the one in eq. (6), this can be rephrased in terms of the control variables (i.e. the DOFs) as follows:

$$\sum_{l=1}^{L^{(1)}} C_{l,j}^{(1)} \cdot d_l^{(1)} + \sum_{l=1}^{L^{(2)}} C_{l,j}^{(2)} \cdot d_l^{(2)} = 0, \quad j = 1, \ldots, n, \tag{17}$$

where $d_l^{(i)}$ are the control variables for patch i ($l = 1, \ldots, L^{(i)}$), and $C_{l,j}^{(i)}$ are constants determined by the derivatives of the basis functions in the j^{th} collocation point. These derivatives on the interface are determined only by the control variables on and around the interface, as indicated in bold in fig. 3b. Denoting these control variables as $\mathbf{d}_{\text{O}}^{(i)}$ (*On* the interface) and $\mathbf{d}_{\text{N}}^{(i)}$ (*Next* to the interface), the constraints in eq. (17) can be written in matrix form as follows:

$$\mathsf{C}_{\text{O}}^{(1)}\mathbf{d}_{\text{O}}^{(1)} + \mathsf{C}_{\text{N}}^{(1)}\mathbf{d}_{\text{N}}^{(1)} + \mathsf{C}_{\text{O}}^{(2)}\mathbf{d}_{\text{O}}^{(2)} + \mathsf{C}_{\text{N}}^{(2)}\mathbf{d}_{\text{N}}^{(2)} = \mathbf{0}, \tag{18}$$

where the j^{th} row of the matrices $\mathsf{C}_{\star}^{(i)}$ contains the corresponding constants $C_{l,j}^{(i)}$ from eq. (17).

Imposing eq. (18) enforces continuity of the gradient of the solution in the case of conforming meshes. For non-conforming meshes, the same master-slave approach with virtual refinement as presented in section 3.1 can be used. Equation (18) then should be expressed for the virtually refined configuration (cf. fig. 2). Defining a master and a slave patch and a set of extraction operators $\mathsf{A}_{\bullet\star}$ such that $\mathbf{d}_{\bullet\star}^{v} = \mathsf{A}_{\bullet\star}\mathbf{d}_{\bullet\star}$ (cf. eq. (11)), it can be expressed as function of the original DOFs:

$$\mathsf{C}_{\text{sO}}\mathsf{A}_{\text{sO}}\mathbf{d}_{\text{sO}} + \mathsf{C}_{\text{sN}}\mathsf{A}_{\text{sN}}\mathbf{d}_{\text{sN}} + \mathsf{C}_{\text{mO}}\mathsf{A}_{\text{mO}}\mathbf{d}_{\text{mO}} + \mathsf{C}_{\text{mN}}\mathsf{A}_{\text{mN}}\mathbf{d}_{\text{mN}} = \mathbf{0}. \tag{19}$$

Expressing this, again using a pseudo-inverse, as function of the slave variables next to the interface and combining it with the C^0-constraint (13) then yields the set of constraints for enforcing C^0- and C^1-continuity:

$$\begin{cases} \mathbf{d}_{\text{sO}} = \mathsf{A}_{\text{sO}}^{+}\mathsf{A}_{\text{mO}}\mathbf{d}_{\text{mO}} \\ \mathbf{d}_{\text{sN}} = -\left(\mathsf{C}_{\text{sN}}\mathsf{A}_{\text{sN}}\right)^{+}\left[\left(\mathsf{C}_{\text{sO}}\mathsf{A}_{\text{sO}}\mathsf{A}_{\text{sO}}^{+} + \mathsf{C}_{\text{mO}}\right)\mathsf{A}_{\text{mO}}\mathbf{d}_{\text{mO}} + \mathsf{C}_{\text{mN}}\mathsf{A}_{\text{mN}}\mathbf{d}_{\text{mN}}\right] \end{cases}. \tag{20}$$

Non-planar shells

For the sake of simplicity, the procedure explained above treated planar shells, and the C^1-coupling was done assuming there are only out-of-plane DOFs. If the shell is instead curved, or contains kinks, the concept of the coupling remains the same but care should be taken that it is imposed to the correct DOFs. To be precise, the gradient of the out-of-plane deformations should be coupled. This can be done by using the unit vectors $\boldsymbol{n}_{\text{patch}}$ normal to the patch surface, on the interface (i.e. the out-of-plane unit vector on the interface). Equation (14) then needs to be modified as follows:

$$\left\langle \left\{ \frac{\partial u_1^{(1)}}{\partial n_{\text{int}}^{(1)}}, \frac{\partial u_2^{(1)}}{\partial n_{\text{int}}^{(1)}}, \frac{\partial u_3^{(1)}}{\partial n_{\text{int}}^{(1)}} \right\}, \boldsymbol{n}_{\text{patch}}^{(1)} \right\rangle + \left\langle \left\{ \frac{\partial u_1^{(2)}}{\partial n_{\text{int}}^{(2)}}, \frac{\partial u_2^{(2)}}{\partial n_{\text{int}}^{(2)}}, \frac{\partial u_3^{(2)}}{\partial n_{\text{int}}^{(2)}} \right\}, \boldsymbol{n}_{\text{patch}}^{(2)} \right\rangle = 0, \tag{21}$$

with $\langle \bullet, \star \rangle$ denoting the dot product and with the dependency on $(\xi, 1)$ left out for the sake of brevity. Starting from this C^1-constraint and using the DOFs u_k in all three coordinate directions X_k, the rest of the coupling procedure proceeds completely analogously. Equation (20) becomes:

$$\begin{cases} \mathbf{d}_{k,\text{sO}} = \mathsf{A}_{\text{sO}}^{+}\mathsf{A}_{\text{mO}}\mathbf{d}_{k,\text{mO}}, \qquad k = 1, 2, 3 \\ \\ \displaystyle\sum_{k=1}^{3} \mathbf{d}_{k,\text{sN}} = -\sum_{k=1}^{3} \left(\mathsf{C}_{k,\text{sN}}\mathsf{A}_{\text{sN}}\right)^{+}\left[\left(\mathsf{C}_{k,\text{sO}}\mathsf{A}_{\text{sO}}\mathsf{A}_{\text{sO}}^{+} + \mathsf{C}_{k,\text{mO}}\right)\mathsf{A}_{\text{mO}}\mathbf{d}_{k,\text{mO}} + \mathsf{C}_{k,\text{mN}}\mathsf{A}_{\text{mN}}\mathbf{d}_{k,\text{mN}}\right] \end{cases}. \tag{22}$$

The constraint matrices $\mathsf{C}_{k,\bullet\star}$ now take into account the components of the normal vector $\boldsymbol{n}_{\text{patch}}$ in coordinate direction k. It is important that the definition of $\boldsymbol{n}_{\text{patch}}$ is consistent across patches, in the sense that it should always originate from the same side of the geometry.

Patch configurations with multiple interfaces

The coupling equations (22) for the entire shell structure (containing one or more interfaces) can be combined into a single matrix equation $\mathsf{R}\mathbf{d}_{\text{ms}} = \mathbf{0}$, where R is the global constraint matrix and $\mathbf{d}_{\text{ms}}$ the vector of all the unknowns involved in the coupling constraints. This vector is the union of all the master and slave DOFs: $\mathbf{d}_{\text{ms}} = \mathbf{d}_{\text{m}} \cup \mathbf{d}_{\text{s}}$. In a general case with more than one interface, the global constraint matrix R might contain some linear dependencies (e.g. due to circular dependencies at patch cross points). These can be removed by computing the Reduced Row Echelon Form (RREF) of R and retaining only the non-zero rows. Assume, without loss of generality, that the DOFs are ordered in such a way that the slave DOFs $\mathbf{d}_{\text{s}}$ act as pivots in the resulting RREF matrix R_{RREF}. The n_{D} remaining linearly independent coupling equations can be expressed as $\mathsf{R}_{\text{RREF}}\mathbf{d}_{\text{ms}} = \mathbf{0}$ with $\mathsf{R}_{\text{RREF}} = \begin{bmatrix}\mathsf{I}_{n_{\text{D}}\times n_{\text{D}}} & -\mathsf{T}_{\text{DI}}\end{bmatrix}$, where $\mathsf{I}_{n_{\text{D}}\times n_{\text{D}}}$ is the identity matrix of size n_{D}. The matrix T_{DI} expresses the resulting relationships that impose the interface coupling between the n_{D} dependent DOFs $\mathbf{d}_{\text{D}}$ (i.e. a subset of the slave DOFs $\mathbf{d}_{\text{s}}$) and the remaining independent DOFs $\mathbf{d}_{\text{I}} = \mathbf{d}_{\text{ms}} \setminus \mathbf{d}_{\text{D}}$:

$$\mathbf{d}_{\text{D}} = \mathsf{T}_{\text{DI}}\mathbf{d}_{\text{I}}. \tag{23}$$

By condensing out the DOFs $\mathbf{d}_{\text{D}}$ from the system matrix according to eq. (23), the global system of equations can now be solved while taking into account the continuity conditions for all the interfaces.

4 Numerical examples

This section demonstrates the accuracy of the proposed coupling method through some numerical case studies. The IGA models are all computed in MATLAB with in-house code that uses functionalities from the IGAFEM framework [3].

4.1 Free vibrations of a hemispherical shell

In [27], Niordson studies the free vibration spectra of thin elastic spherical shells, based on Kirchhoff-Love shell theory. He finds that bending modes constitute an important part of such spectra, in particular in the lower frequency region. Therefore, the bending behaviour of such shells should be properly incorporated in order to accurately model the vibration behaviour. An accurate C^1-coupling method is therefore crucial here.

Consider a hemispherical shell of radius R, which can be modelled geometrically exactly using a multipatch NURBS discretisation of fourth order. This is illustrated by the mesh in Figure 4, comprising four identically sized patches along the base of the sphere, and one bigger patch on the top. Niordson gives reference values for the free-free eigenfrequencies, in the form of the dimensionless number Ω [27]:

$$\Omega = 2\pi f R\sqrt{(1-\nu^2)\frac{\rho}{E}}, \tag{24}$$

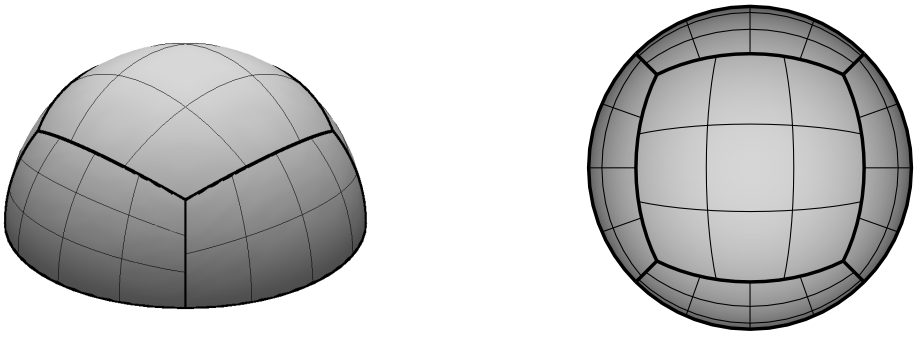
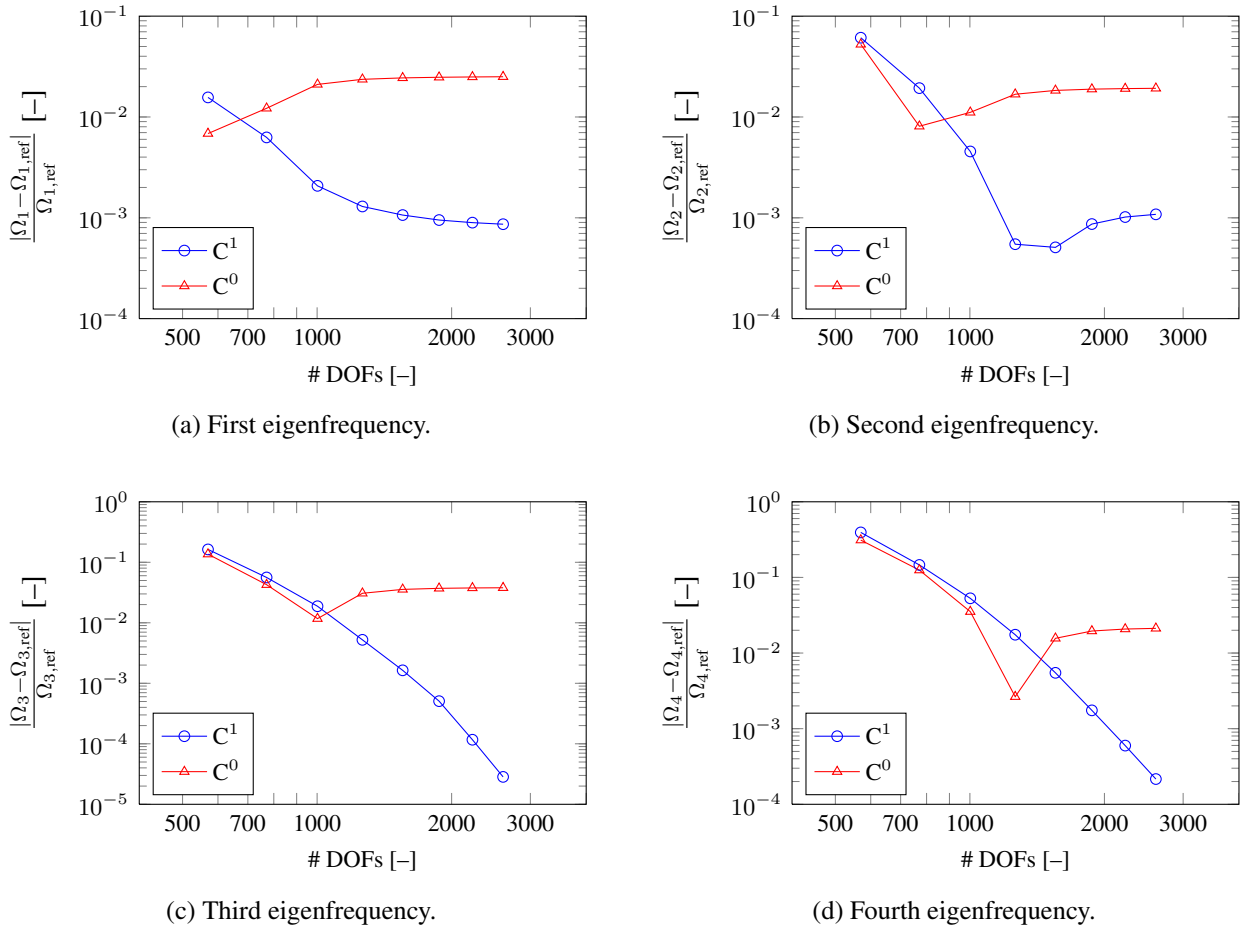


Figure 4: Non-conforming NURBS mesh of the hemisphere, consisting of five biquartic patches. The thick lines indicate the patch boundaries.

where f is the eigenfrequency. To evaluate the accuracy of the proposed patch coupling method, the eigenfrequencies are studied numerically using non-conforming meshes (like in fig. 4) and compared to Niordson's reference values. The numerical values used here are $R = 1\,\text{m}$, $t = R/100$, $E = 200\,\text{GPa}$, $\rho = 7800\,\text{kg/m}^3$, $\nu = 0.3$. Figure 5 plots the convergence of the first four non-zero eigenfrequencies, comparing the enforcing of C^1-continuity versus only imposing C^0-continuity. The plots clearly show that C^0-continuity alone is not sufficient to ensure convergence. For the first two eigenfrequencies it seems as if also the accuracy using the full C^1-coupling stagnates, but this is due to the limited number of significant digits for the small corresponding reference values given in [27] (respectively $\Omega_{1,\text{ref}} = 0.0123$ and $\Omega_{2,\text{ref}} = 0.0339$). The third and fourth eigenfrequencies show that the C^1-coupling does converge.



(a) First eigenfrequency.

(b) Second eigenfrequency.

(c) Third eigenfrequency.

(d) Fourth eigenfrequency.

Figure 5: Convergence of the relative error for the four lowest non-zero eigenfrequencies of a hemispherical shell, comparing C^1- versus C^0-continuity enforcement. Without imposing the C^1-continuity, there is no convergence.

4.2 Forced response of a structure containing kinks

This example studies a curved structure containing kinks which is clamped at its bottom edges, as shown in fig. 6. It consists of six B-spline patches: the four patches making up the top part have curved interfaces and the clamped patches in addition meet their neighbours at a kink (indicated in green), i.e. a line where there is no G^1-continuity (continuity of the gradient of the geometry mapping). This in order to present a general patch coupling configuration. The structure is excited by a point force of 1 kN in the negative z-direction placed at the centre of the outer edge. Since this point force as well as the boundary counditions are placed symmetrically, the response should also be symmetric (with respect to the x-axis).

The dynamic response of this shell configuration is studied in a frequency range from 0 to 500 Hz, in particular in the response point indicated by the blue dot in fig. 6. An FE model is used as reference solution,

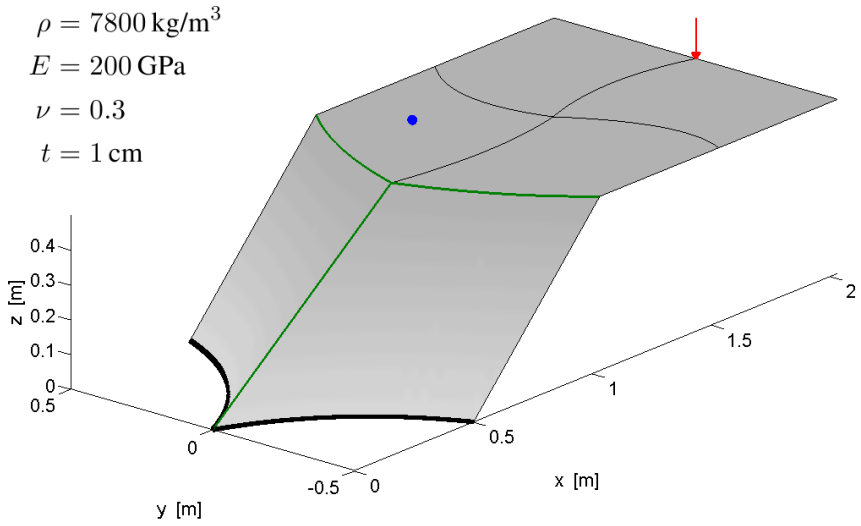
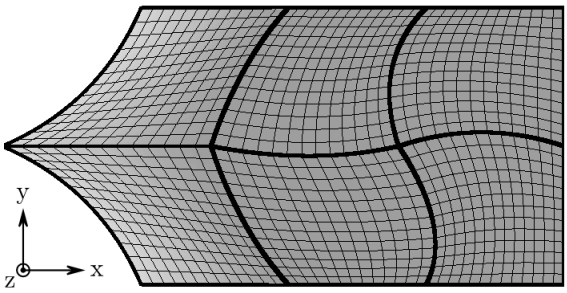
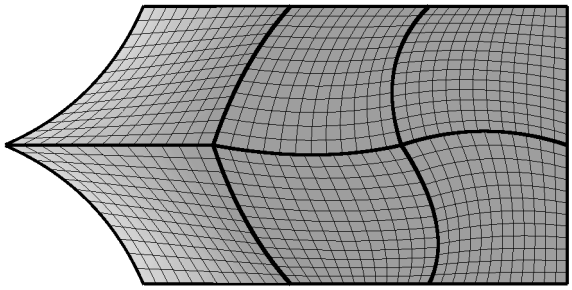


Figure 6: Kinked shell structure consisting of 6 patches (kinks indicated in green). It is clamped at the bottom edges (indicated by the thick black lines) and excited by a point force at the centre point of the outer boundary (indicated by a red arrow). The blue dot is the response point for analysing the frequency behaviour.

computed using 20 801 `SHELL63` elements in ANSYS. With 6 DOFs per node, this leads to more than 126 000 reference DOFs. Using a 10 elements per wavelength rule of thumb, this reference model should be valid up to 2500 Hz. Figure 7 shows the IGA meshes used for verification against this reference model. They are bicubic B-spline patches, both a conforming and a non-conforming one. The meshes comprise 1014 and 939 elements, respectively. This respectively results in 4608 and 4329 uncoupled DOFs. After eliminating the slave DOFs using the coupling equations, the models are left with 4168 and 3887 master variables to be solved for, respectively.



(a) Conforming mesh (4608 DOFs).



(b) Non-conforming mesh (4329 DOFs).

Figure 7: The bicubic B-spline meshes used for the isogeometric shell analysis of the kinked structure (top view, looking in the negative z-direction). The thick lines indicate the patch boundaries.

Figure 8 plots the frequency response curves for the vertical displacement of the response point (cf. fig. 6) for these two meshes. Both curves show good agreement with the reference solution. The accuracy decreases with increasing frequency, where a very slight shift of the response is introduced. This is the onset of the well-known pollution effect [28]. Interesting to note is that there is no significant difference in performance between the conforming and the non-conforming meshes.

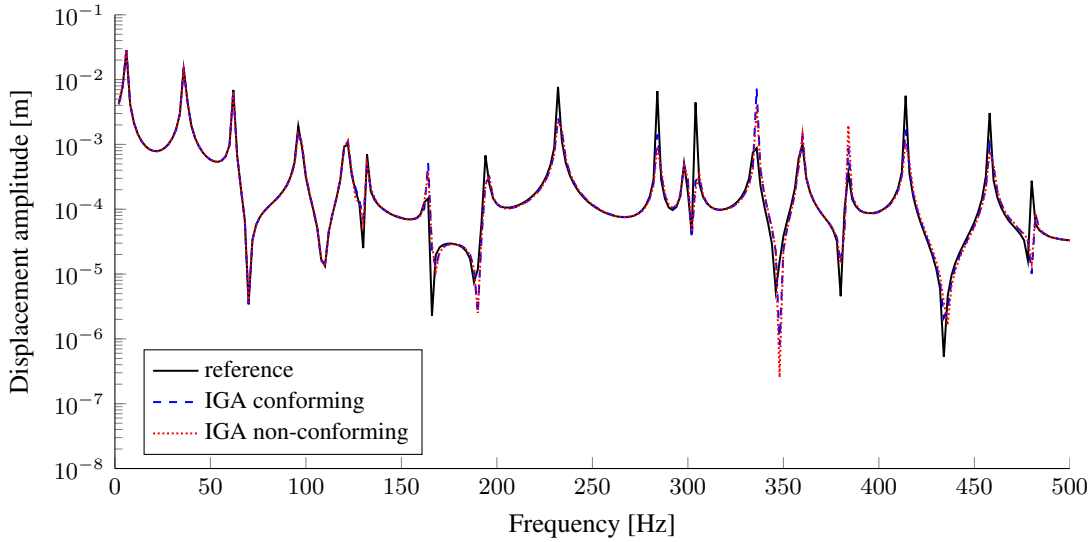


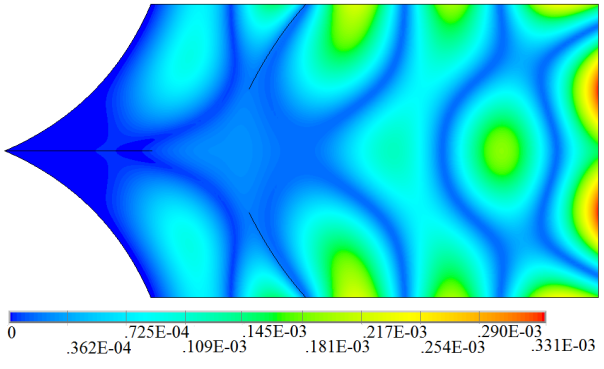
Figure 8: Amplitude of the vertical displacement in the response point in a frequency range from 0 to 500 Hz (with a frequency resolution of 2 Hz), for the conforming and the non-conforming IGA model.

To assess the global accuracy of the IGA solutions, fig. 9 compares two plots of the total displacement amplitude obtained using the non-conforming model (cf. fig. 7b) and compares them to the FEM reference solution. Although the colour bars are slightly different between MATLAB and ANSYS, the correct deformation fields can be clearly recognised in figs. 9b and 9d. Note that even though the IGA mesh is non-symmetric (fig. 7b), the solutions are fully symmetric with respect to the x-axis (as they should be in this case).

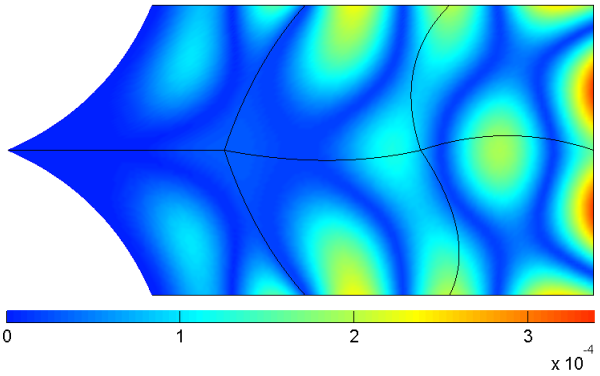
5 Conclusions and outlook

A coupling method is introduced to enforce C^1-continuity across NURBS patches in isogeometric Kirchhoff-Love shell analysis. Although both the C^0- and the C^1-parts of the coupling make use of a master-slave formulation in the case of non-conforming patches, the approaches for the two parts are quite different. The C^0-coupling consists of imposing constraints that hold for every (parametric) location along the interface, whereas the C^1-coupling explicitly matches first-order derivatives of the discretisation in certain well-chosen collocation points. More specifically, the coupling imposes the gradient of the (out-of-plane) deformation field in the (in-plane) direction normal to the interface to be equal across the interface. Based on these C^0- and C^1-constraints and on the choice for master and slave patches (where the coarser patch is chosen as the master), the slave variables can be eliminated from the system of equations, resulting in a reduced system to be solved. The coupling method is assessed using two numerical case studies of dynamic shell analysis, including curved geometries as well as kink connections. The coupling is found to yield accurate results, also for non-conforming patch configurations where even a C^0-coupling is non-trivial.

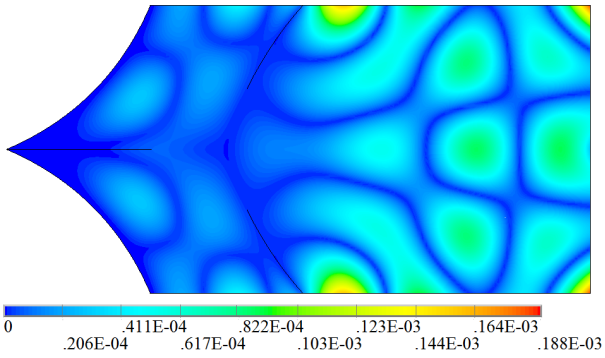
Future work will further extend the verification of the coupling method, including more complex geometries and more general patch configurations. Of particular interest are configurations containing partially overlapping interfaces (as opposed to full-length interfaces) as well as trimmed geometries.



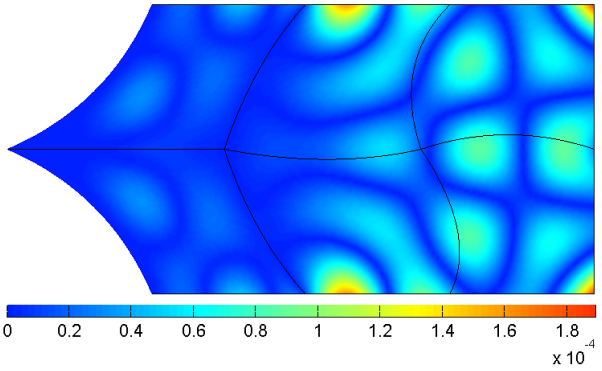
(a) FEM reference solution at 250 Hz.



(b) IGA solution at 250 Hz.



(c) FEM reference solution at 500 Hz.



(d) IGA solution at 500 Hz.

Figure 9: Top view of the plots of the total displacement amplitude [m] at 250 and 500 Hz for the non-conforming IGA model (with indication of the patch boundaries) compared to the FEM reference solution.

Acknowledgements

The research of Laurens Coox is funded by a PhD grant of the agency of Innovation by Science and Technology in Flanders (IWT Flanders). The IWT Flanders is also gratefully acknowledged for its support within the OPTIWIND project. The European Commission is gratefully acknowledged for its support of the ANTARES research project (GA606817). The presented work is also partially supported by the Research Fund KU Leuven and by Flanders Make, the strategic research centre for the manufacturing industry.

References

[1] T. J. R. Hughes, J. Cottrell, Y. Bazilevs, *Isogeometric analysis: CAD, finite elemens, NURBS, exact geometry and mesh refinement*, Computer Methods in Applied Mechanics and Engineering, Vol. 194, (2005), pp. 4135–4195.

[2] J. Cottrell, T. Hughes, Y. Bazilevs, *Isogeometric Analysis: Toward Integration of CAD and FEA*, John Wiley & Sons (2009).

[3] V. P. Nguyen, C. Anitescu, S. P. Bordas, T. Rabczuk, *Isogeometric analysis: an overview and computer implementation aspects*, Mathematics and Computers in Simulation, Vol. 117, (2015), pp. 89–116.

[4] G. M. Stanley, *Continuum-based shell elements*, Ph.D. thesis, Division of Applied Mechanics, Stanford University (1985).

[5] J. Cottrell, T. J. R. Hughes, A. Reali, *Studies of refinement and continuity in isogeometric structural analysis*, Computer Methods in Applied Mechanics and Engineering, Vol. 196, (2007), pp. 4160–4183.

[6] S. K. Kleiss, C. Pechstein, B. Jüttler, S. Tomar, *IETI – Isogeometric Tearing and Interconnecting*, Computer Methods in Applied Mechanics and Engineering, Vol. 247–248, (2012), pp. 201–215.

[7] M. Ruess, D. Schillinger, A. Özcan, E. Rank, *Weak coupling for isogeometric analysis of non-matching and trimmed multi-patch geometries*, Computer Methods in Applied Mechanics and Engineering, Vol. 269, (2014), pp. 46–71.

[8] A. Apostolatos, R. Schmidt, R. Wüchner, K.-U. Bletzinger, *A Nitsche-type formulation and comparison of the most common domain decomposition methods in isogeometric analysis*, International Journal for Numerical Methods in Engineering, Vol. 97, (2014), pp. 473–504.

[9] E. Brivadis, A. Buffa, B. Wohlmuth, L. Wunderlich, *Isogeometric mortar methods*, Computer Methods in Applied Mechanics and Engineering, Vol. 284, (2015), pp. 292–319.

[10] W. Dornisch, G. Vitucci, S. Klinkel, *The weak substitution method — an application of the mortar method for patch coupling in NURBS-based isogeometric analysis*, International Journal for Numerical Methods in Engineering, Vol. 103, (2015), pp. 205–234.

[11] J. Kiendl, K.-U. Bletzinger, J. Linhard, R. Wüchner, *Isogeometric shell analysis with Kirchhoff-Love elements*, Computer Methods in Applied Mechanics and Engineering, Vol. 198, (2009), pp. 3902–3914.

[12] J. Kiendl, Y. Bazilevs, M.-C. Hsu, R. Wüchner, K.-U. Bletzinger, *The bending strip method for isogeometric analysis of Kirchhoff-Love shell structures comprised of multiple patches*, Computer Methods in Applied Mechanics and Engineering, Vol. 199, (2010), pp. 2403–2416.

[13] R. Schmidt, R. Wüchner, K.-U. Bletzinger, *Isogeometric analysis of trimmed NURBS geometries*, Computer Methods in Applied Mechanics and Engineering, Vol. 241–244, (2012), pp. 93–111.

[14] S. Shojaee, E. Izadpanah, N. Valizadeh, J. Kiendl, *Free vibration analysis of thin plates by using a NURBS-based isogeometric approach*, Finite Elements in Analysis and Design, Vol. 61, (2012), pp. 23–34.

[15] F. Maurin, L. Coox, F. Greco, E. Deckers, C. Claeys, W. Desmet, *Bloch-Floquet theorem for isogeometric analysis of periodic problems governed by high-order partial differential equations*, manuscript submitted for publication in *Computer Methods in Applied Mechanics and Engineering*.

[16] L. Coox, F. Greco, O. Atak, D. Vandepitte, W. Desmet, *A robust patch coupling method for NURBS-based isogeometric analysis of non-conforming multipatch surfaces*, Computer Methods in Applied Mechanics and Engineering (2016), *In press.* DOI: http://dx.doi.org/10.1016/j.cma.2016.06.022.

[17] L. Coox, O. Atak, D. Vandepitte, W. Desmet, *An isogeometric indirect boundary element method for solving acoustic problems in open-boundary domains*, Computer Methods in Applied Mechanics and Engineering (2016), *In press.* DOI: http://dx.doi.org/10.1016/j.cma.2016.05.039.

[18] M. Breitenberger, A. Apostolatos, B. Philipp, R. Wüchner, K.-U. Bletzinger, *Analysis in computer aided design: Nonlinear isogeometric B-Rep analysis of shell structures*, Computer Methods in Applied Mechanics and Engineering, Vol. 284, (2015), pp. 401–457.

[19] L. Piegl, W. Tiller, *The NURBS Book*, Springer-Verlag Berlin Heidelberg (1997).

[20] M. Cox, *The numerical evaluation of B-splines*, DNAC 4, National Physics Laboratory (1971).

[21] C. de Boor, *On calculation with B-splines*, Journal of Approximation Theory, Vol. 6, (1972), pp. 50–62.

[22] R. Craig, A. Kurdila, *Fundamentals of Structural Dynamics*, John Wiley & Sons, Inc. (2006).

[23] M. J. Borden, M. A. Scott, J. A. Evans, T. J. Hughes, *Isogeometric finite element data structures based on Bézier extraction of NURBS*, International Journal for Numerical Methods in Engineering, Vol. 87, (2011), pp. 15–47.

[24] C. de Boor, *A practical guide to splines*, Springer-Verlag (1979).

[25] F. Auricchio, L. Beir ao da Veiga, T. Hughes, A. Reali, G. Sangalli, *Isogeometric collocation methods*, Mathematical Models and Methods in Applied Sciences, Vol. 10, (2010), pp. 2075–2107.

[26] S. Demko, *On the existence of interpolation projectors onto spline spaces*, Journal of Approximation Theory, Vol. 43, (1985), pp. 151–156.

[27] F. I. Niordson, *The spectrum of free vibrations of a thin elastic spherical shell*, International Journal of Solids and Structures, Vol. 24, (1988), pp. 947–961.

[28] P. Bouillard, F. Ihlenburg, *Error estimation and adaptivity for the finite element method in acoustics: 2D and 3D applications*, Computer Methods in Applied Mechanics and Engineering, Vol. 176, (1999), pp. 147–163.